REMARKS

The Office Action, mailed June 25, 2008, considered and rejected claims 1-16. Claims 1-16 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Baisley* (U.S. Patent No. 6,106,574) in view of *Haikin* (U.S. Patent No. 6,757,893).[2]

By this paper, claims 1 and 11 are amended, claims 5 and 16 are cancelled, and claim 17 is added, such that following this paper, claims 1-4, 6-15 and 17 are pending, of which claims 1 and 11 are the only independent claims at issue.

As reflected in the claims listed above, Applicant's claims relate to storage media and systems for associating original source code with binary code to allow the binary code to be debugged. As reflected in claim 1, for example, a source code file that includes source code and an associated version are stored on a server in a source code file. The source code file is compiled into a binary file and, while doing so, information identifying the location of the source code file and the version of the source code file is extracted. Also extracted at that time is a name of the server, a port of the server at which the server may be accessed to access the source code, a path to the source code, and a numeric value indicating a version number of the source code. The extracted information is stored in a debug file associated with the binary file. After compiling, an instruction can then be received to debug the binary file. Following such, the extracted information is used from the debug file and the source code file is located and associated with the binary file. Thereafter, the debugging is performed on the binary file with full source code support by correlating lines of the source code with binary instructions in the binary file, and such that the source code file includes only the source code originally used to compute the binary file. Claim 11 recites a system generally capable of performing a method similar to that performed upon executing the instructions contained in the computer-readable storage media of claim 1.

While the cited art generally relates to relating objects in a compiler to locations in a source program and/or to version control systems, Applicant respectfully submits that the cited art fails to disclose or reasonably support each of the elements of the pending claims. For example, among other things, the cited art fails to disclose or reasonably support that while a source code file is being debugged, the version associated with the source code, a name of the server, a port of the server at which the server may be accessed to get access to the source code, a path to the source code, and a

[2] Although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

numeric value indicating a version number of the source code, are each extracted and stored in a debug file, as such is recited in combination with the other claim elements.

For example, *Baisley* generally describes an object-oriented method for relating objects in a compiler to locations in a source program and to inlined call histories. More particularly, once or more source files are provided to a compiler which parses them to generate objects that represent program elements. Those objects are then converted by the compiler into machine instructions that are stored in a binary file. That file also includes mapping instructions to source locations. (Col. 5, ll. 12-22).

The compiler, must relate the objects to locations within the source files themselves. At the time the source files are parsed, the compiler creates objects representing program elements, and a source location of the objects is shown as a source file name and a line number within the source file. As the machine instructions are created, the machine instruction object is generally given the same source location. (Col. 5, ll. 23-35).

The source location of these objects can be used in two cases. In the first, the compiler shows the source location when issuing a diagnostic message to inform the compiler's user of an error location. In a second case, the compiler puts a table in the binary file to map the instructions to their corresponding source locations. The table is also used for debugging when the machine instructions are loaded from the binary file for execution. When processing is then interrupted, the debugger can use the table to find the source location relating to the current point of instructions. (Col. 5, ll. 36-51). To fully operate, the system requires the compiler to use the type Src which identifies a source location, a list of inlined source locations. An object of this type has a single instance variable, which is an integer. (Col. 5, ll. 52-55; Col. 6, ll. 15-21).

An object type, SrcRange also is used to include information about a range of source lines, and includes six variables that contain: the first and last index for a range, an indicator of whether a simple source range or compound inline range is used, the source file name, object file name, first line number for the range, source index for a call, source index for a line, and/or an inlined function identifier. (Col. 6, ll. 49-67). Other object types SrcFineThl and SrcTblRange may also be used to contain information generally tracking most recently accessed objects and/or pointers to fine or course tables. (Col. 7, ll. 32-67).

Accordingly, while *Baisley* discloses generally identifying a source code file by its name and potentially location, it can be easily seen that it fails to disclose any identification of at least a port of the server at which the server may be accessed to access the source code. Indeed, as acknowledged by the Office, *Baisley* fails to disclose the storage of source code on a server and, as such that there would be no need to include a port number for how to access the source code on a server.

Applicant also respectfully submits that *Haikin* fails to remedy the deficiencies of *Baisley*. In particular, *Haikin* generally relates to a software source code version control system during development and maintenance of software by multiple software developers, and allows version tracking on a line-by-line basis. The Office Action notes that a workstation in *Haikin* makes a request to access source code from a server through a LAN, and thus concludes that a port of the server is inherent. (Office Action, op. 8). Applicant respectfully submits that even if it is inherent that a port of a server is inherent in the communications in *Haikin*, such communication does not render the claims unpatentable. For example, merely communicating through a port is not what is recited in the claims. Rather, the claims recite that while compiling the source code file, information is extracted that identifies the location, path, version number, and name of the source code file, as well as the name and port of the server. The claims thus require that the port be extracted while compiling the source code and prior to receiving instructions for debugging. In contrast, the Office appears to merely argue that it is inherent that communication is through a port. In other words, even if the Office's assertion is treated as true, it merely means that when the workstation and server communicate, they do so through a port, but it does not make it inherent that while compiling source code, a port of the server is extracted. Indeed, the Office clearly states that it is inherent that the port be used when the access to the source code is requested, such that there would be no need to extract the port at compile time as it would then already have been necessarily present.

Additionally, Applicant further notes that the assertion of inherency, fails to satisfy the requirements set forth by the Office. In particular, a rejection based on inherency cannot stand if the result or characteristic <u>may</u> occur or be present, but instead must be "necessarily present" in the thing described in the reference. (M.P.E.P. § 2112). Furthermore, in relying on a theory of inherency, the Office "must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic <u>necessarily</u> flows from the teachings of the applied prior art." (*Id.*). Only after such a showing by the Office has been made can the burden of disproving inherency be placed on the Applicant.

In the present application, the Office merely states that "a port of server is inherent." (Office Action, p. 8). Accordingly, the Office has merely concluded that a port of a server is inherent, but has provided no basis in fact or technical reasoning as required. Mere conclusions, without the required factual and/or technical basis, cannot satisfy the burden on the Office. Moreover, as discussed above, such a conclusion also cannot possibly satisfy the claim elements as written, in which the port is extracted while compiling, and with the other information recited, in combination with the other claim elements.

In view of the foregoing, Applicant respectfully submits that the other rejections to the claims are now moot and do not, therefore, need to be addressed individually at this time. It will be appreciated, however, that this should not be construed as Applicant acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any official notice. Instead, Applicant reserves the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicant specifically requests that the Examiner provide references supporting the teachings officially noticed, as well as the required motivation or suggestion to combine the relied upon notice with the other art of record.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney at (801) 533-9800.

Dated this 30th day of October, 2008.

Respectfully submitted,

/Colby C. Nuttall, Reg. # 58146/

RICK D. NYDEGGER
Registration No. 28,651
COLBY C. NUTTALL
Registration No. 58,146
Attorneys for Applicant
Customer No. 047973

RDN:CCN:gd
TEMPLATE FOR AMENDMENT B AFTER FINAL OA 25JUNE2008 DOC.DOC